

Harness the Nature for Computation

Yasuhiro Suzuki

Department of Complex Systems Science, Graduate School of Information Science,
Nagoya University, Furocho Chikusa Nagoya City 464-8601, Japan
ysuzuki@nagoya-u.jp

Abstract. *Natural computing* investigates and models computational techniques inspired by nature and attempts to understand natural phenomena as information processing. In this position paper, we consider harness the nature for computation, from the perspective of natural computing. We investigated facsimile computational models of self-organization in nature, and identified dissipation of information flow as a common mechanism, where intermediate information is produced through interactions and consumed through evoking novel interactions. Based on this mechanism, we propose the concept of a harness: an indirect controlling method for natural systems. We realize this concept through a computational model, and discuss how this concept has already been successfully applied in medical and ecological science.

Keywords: Harness, Dissipation of Information, Self-organization.

1 Introduction

The principle of natural computing lies in understanding nature in terms of computing[10]. In order to understand a natural phenomenon, we must define all related elements and interactions precisely; otherwise, we shall not be able to obtain the algorithm for the phenomenon. However, it is virtually impossible for humans to amass all the knowledge required for such purposes.

Although computing has been used in activities such as sheep husbandry, it is not possible to know everything about sheep, e.g., the manner in which they communicate. It is known that sheep are flock animals with a natural inclination to follow a leader; therefore, a shepherd or sheepdog can easily control a flock of sheep by intimidation. Shepherds adopt strategic intimidation with a specific objective; we refer to such a strategy as a “harnessing.” A harness is a sequence of instructions that can be modified to alter sheep behaviour. Hence, we can design a harness for guiding a flock by changing the order of instructions. Therefore, a harness can be regarded as an algorithm for computation. However, it is different from a conventional algorithm, which requires precise knowledge of all the related elements and interactions. Using a harness, a shepherd can control a flock of sheep without knowing all about the sheep and their interactions.

To design a harness, we must have precise information about the instructions to be used; the construction of a harness is similar to that of a computer algorithm. When the designed harness is applied to natural systems, it may have

desired as well as undesired effects, thereby altering the system behaviour. Such effects and the resulting changes in behaviour may be unexpected.

Hence, computation using a harness differs from that using an algorithm. The execution of sequential instructions is similar in both cases; however, in the case of a harness, the computation result does not solve the problem directly, but produces effects and side effects in the system and guides the system to solve the problem.

In this paper, we consider the use of a harness for natural computing by investigating an algorithm for spontaneous self-organization in nature, on the basis of an artificial chemistry known as abstract rewriting system on multisets (ARMS).

2 Abstract Rewriting System on Multisets, ARMS

An Abstract Rewriting System on Multisets (ARMS, [14]) is a model of artificial chemistry [1] based on computational algebra (rewriting systems) and physical chemistry (such as Gillespie's method [5]). We computationally characterized the Edge of Chaos [15] by investigating the relationship between ARMS and a cellular automaton. We showed how (computational) living things emerge through chemical evolution and can be evolved using an ARMS, by applying this evolutionary system to solve a simple mathematical problem [16]. Furthermore, we propose a model of evolutionary dynamics for the proto-enzyme through an evolutionary reaction network modeled by an ARMS, where repeated auto-catalytic reaction networks emerge and are catastrophically destroyed [19]. This type of behavior has also been reported using a replicator system [2]. Beyond the field of artificial life, ARMS has also been used in ecology [18], medical science [21] and environment engineering [6] among others.

ARMS is calculated as an expression of expressed of the Chemical Master Equation (CME), a stochastic expression of the Reaction Rate Equation (RRE). Here, we demonstrate that an ARMS can be regarded as a CME and, through continuous approximation, the deterministic RRE, which is denoted by a set of ordinal differential equations that can be obtained from an ARMS [17].

Fundamentally, an ARMS is a construct $\Gamma = (A, w, R)$, where A is an *alphabet*, w is a multiset present in the initial configuration of the system, and R is the set of multiset rewriting rules. Let A be an alphabet (a finite set of abstract symbols). A multiset over A is a mapping $M : A \mapsto \mathbf{N}$, where \mathbf{N} is the set of natural numbers; $0, 1, 2, \dots$. For each $a_i \in A$, $M(a_i)$ is the multiplicity of a_i in M . We can also denote $M(a_i)$ as $[a_i]$. We denote by $A^\#$ the set of all multisets over A , with the empty multiset, \emptyset , defined by $\emptyset(a) = 0$ for all $a \in A$.

A multiset $M : A \mapsto \mathbf{N}$, for $A = \{a_1, \dots, a_n\}$ is represented by the state vector $w = (M(a_1), M(a_2), \dots, M(a_n))$, w . The union of two multisets $M_1, M_2 : A \mapsto \mathbf{N}$ is the addition of vectors w_1 and w_2 , representing the multisets M_1 and M_2 , respectively. If $M_1(a) \leq M_2(a)$ for all $a \in A$, then we say that multiset M_1 is included in multiset M_2 and write $M_1 \subseteq M_2$. A rewriting rule r over A can be defined as two multisets, (s, u) , with $s, u \in A^\#$. A set of rewriting rules is

expressed as R . A rule $r = (s, u)$ can also be represented as $r = s \rightarrow u$. Given a multiset $w \subseteq s$, the application of a rule $r = s \rightarrow u$ to the multiset w produces a multiset w' such that $w' = w - s + u$. Note that s and u can also be zero vector (i.e., empty).

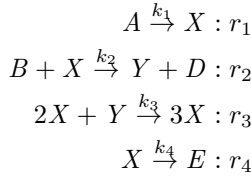
3 Simulation of Natural Systems

We can model multiscale self-organizing systems such as chemical reactions, membrane systems (protocell models)[13], signal transduction systems in cells, and ecosystems using ARMS.

Belousov-Zhabotinskii Reaction

The Belousov-Zhabotinskii (BZ) reaction displays a remarkable repertoire of complex behavior, including periodic and chaotic temporal oscillations, multiple stable stationary states, temporally and spatially periodic expanding target patterns, and rotating multi-armed spiral waves [3].

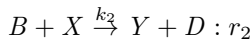
A simple abstract chemical scheme of the BZ reaction has been proposed by Prigogine and colleagues [11] in the form of the following rules:



In a simulation of the Brusselator model, the frequency of applying rewriting rules follows the law of mass action, and the probabilities can be described as follows:

$$\begin{aligned} \text{Prob}(\mathbf{x} \rightarrow \mathbf{x} + \mathbf{r}_1) &= k_1 \\ \text{Prob}(\mathbf{x} \rightarrow \mathbf{x} + \mathbf{r}_2) &= k_2 x \\ \text{Prob}(\mathbf{x} \rightarrow \mathbf{x} + \mathbf{r}_3) &= k_3 x^2 y \\ \text{Prob}(\mathbf{x} \rightarrow \mathbf{x} + \mathbf{r}_4) &= k_4 x \\ \text{Prob}(\mathbf{x} \rightarrow \mathbf{x}) &= 1 - (k_1 + k_2 x + k_3 x^2 y + k_4 x). \end{aligned} \tag{1}$$

The results of these simulations are shown in Figure 1. The model exhibits oscillation between the value of X and Y (the limit cycle), where ARMS agrees well with the kinetics of the differential equation model. In this model,



plays a key role: in order to activate or inhibit this reaction, the intermediate substance Y is required, and so the rates of producing and consuming Y control the system's behaviors.

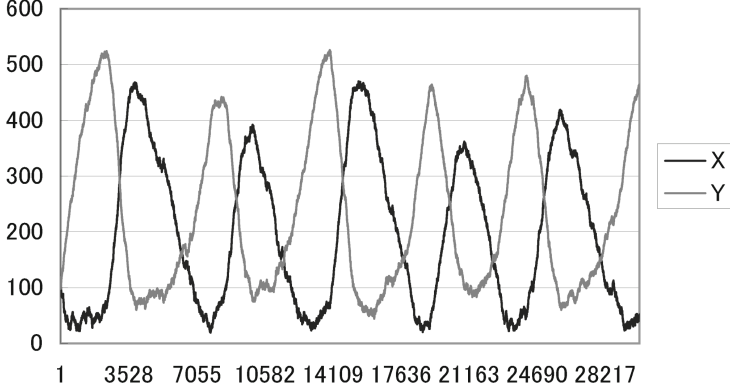
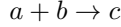


Fig. 1. Limit cycle behavior of ARMS for the Brusselator system. Parameters: $k_1 = 100$, $k_2 = 3$, $k_3 = 10^{-3}$, $k_4 = 1$.

Chemical Autopoiesis

Chemical autopoiesis has been proposed by L. Luisi [8] and F. Varela [9] as representing a model for abiogenesis, and aspects of it have already been chemically realized [8], [23]. Chemical autopoiesis is a chemical system based on the surfaces of water drops in oil, where oil-soluble chemical “a” and water-soluble chemical “b” produce surfactant molecule “c” between water drops and oil; when the concentration of “c” becomes large, a water drop covered by “c” is divided into water drops of random size.



In order to describe the membrane structure in ARMS we define the language MS over the alphabet $\{[,]\}$ whose strings are recurrently defined as follows:

- (1) $[,] \in MS$
- (2) if $\mu_1, \dots, \mu_n \in MC, n \geq 1$ then $\{\mu_1, \dots, \mu_n\} \in MS$
- (3) there is nothing else in MS

The outermost membrane M_0 corresponds to a container, such as a test tube or reactor, that never dissolves. We describe the ARMS with the membrane as an Active Cell System (ACS); a transition of ACS is denoted by the construct:

$$\Gamma = (A, \mu, M_1, \dots, M_n, R, MC, \delta, \sigma),$$

where:

- (1) A is a set of objects;
- (2) μ is a membrane structure (it can be changed throughout a computation);
- (3) M_1, \dots, M_n are multisets associated with the regions $1, 2, \dots, n$ of μ ;
- (4) R is a finite set of multiset evolution rules over A .

- (5) MC is a set of membrane compounds;
- (6) δ is the threshold value of dissolving a membrane;
- (7) σ is the threshold value of dividing a membrane;

μ is a membrane structure of degree n , $n \geq 1$, with the membranes labeled in a one-to-one manner, for instance, with the numbers from 1 to n . In this way, also the regions of μ are identified by the numbers from 1 to n .

- (1) All the rules are applied in parallel. In every step, all the rules are applied to all applicable objects in every membrane. If there are more than two rules that can apply to an object, then one rule is selected randomly.
- (2) If a membrane dissolves, then all the objects in its region are left free in the region immediately above it.
- (3) All objects and membranes not specified in a rule and that do not evolve are passed unchanged to the next step.

Dissolving and Dividing a Membrane

Dissolving a membrane is defined as follow:

$$[{}_ha, \dots [{}_ib, \dots]_i]_h \rightarrow [{}_ha, b, \dots]_h,$$

where the ellipsis $\{\dots\}$ illustrate chemical compounds inside the membrane. Dissolving takes place when

$$\frac{|w_i|_{MC}}{|M_i|} < \sigma$$

where σ is the threshold value for dissolving the membrane. All chemical compounds in its region are then freed and merge into the region immediately above it.

Dissociation of Membrane Compounds

Each compound in a membrane will break off over time, and these dissipated membrane compounds are subsequently merged into the region immediately above it: for example,

$$[{}_0a, b, c, c, c[{}_1c, c, c, a, b]_1]_0 \rightarrow [{}_0a, b, c, c, c, c[{}_1c, c, a, b]_1]_0,$$

where membrane compound c in membrane 1 is dissipated and dissolved into the upper region. Hence in order to maintain a membrane, membrane compounds need to be continuously produced to exceed σ , otherwise the membrane will dissolve.

When the volume of membrane compounds reaches a designated threshold, then the membrane is divided. Membrane division is realized by dividing the multisets into random sizes. The frequency of membrane division is proportionate to its size as the size of a multiset becomes larger, the cell divides more frequently:

$$[{}_ha, b, \dots]_h \rightarrow [{}_ha, \dots [{}_ib, \dots]_i]_h$$

and membrane division takes place when:

$$\frac{\|w_h\|_{MC}}{\|M_h\|} > \sigma$$

where σ is the threshold for dividing the membrane. All chemical compounds in its region are then freed and separated randomly by new membranes.

Evolution of Cells

When a cell grows and reaches the threshold value for dividing, it divides into parts of random sizes. This can be seen as a mutation. If a divided cell does not have any membrane compounds, it must disappear immediately. As such, maintaining the membrane through chemical reactions inside the cell can be seen as natural selection. If a cell cannot maintain its membrane, it must disappear. Thus, both dividing and dissolving membranes produce evolutionary dynamics. Hence, all surviving cells in the ACS fit the conditions for Gnti's chemoton model [4].

Behavior of ACS

We set the ACS as $\Gamma = (A = \{a, b, c\}, \mu = \{[,]_0, \dots [,]_{100}\}, M_0 = \{[a^{10}, b^{10}, c^{10}]^{100}\}, R, MC = \{c\}, \delta = 0.4, \sigma = 0.2)$ where;

- (1) R, the length of the left or right-hand-side of a rule is between one and three. Both sides of the rules are obtained by sampling with replacement of the three symbols a, b, and c;
- (2) Membrane structures are assumed to be ($\mu = \{[]_1, \dots []_{100}\}$).

step	state
0.	$[a^{10}, b^{10}, c^{10}]$
1.	$[a^2, b^5, c^{10}]$
2.	$[a^{10}, b^2, c^7]$
3.	$[a^{11}, b^3, c^7]$
3.	$[a^6, b^4, c^5]$

10.	$[a^7, b^4, c^2]$

16.	$[a^1, b^4]$

Fig. 2. An example of state transition of ACS (λ_e closes to 0.0)

λ_e Parameter

In order to investigate the correlation between the characteristics of the rewriting rule and the behavior of the model, we will introduce the λ_e parameter [15] of ARMS;

$$\lambda_e = \frac{\Sigma r_{\Delta S > 0}}{1 + (\Sigma r_{\Delta S < 0} - 1)},$$

where $\Sigma r_{\Delta S > 0}$ corresponds to the number of *heating rules*, and $\Sigma r_{\Delta S < 0}$ to the *cooling rules*. Heating rules take effect when the right hand side of a given rule is larger than its left hand side, while cooling rules take effect when the left hand side of a rule is larger than its right hand side. Hence, λ_e indicates the degree of reproduction of chemicals inside a cell: when λ_e approaches 0.0, the reproduction of chemicals is small, and when it approaches 1.0, reproduction is large.

step	state
0.	$[a^{10}, b^{10}, c^{10}]$
1.	$[a^{10}, b^{10}, c^{10}]$
.....	
91.	$[a^{10}, b^{10}, c^{10}]$
92.	$[[b^2], [b^3, c^1]]$
93.	$[[b^2], [b^1, c^1]]$
94.	$[a^{10}, b^{10}, c^{10}]$
.....	
114.	$[a^2, b^5]$
115.	$[[a^1, b^2][a^1 b^3]]$
116.	$[[a^1, b^2][b^2][a^1 b^3]]$
.....	
140.	$[[[a^1, b^2][b^2, c^1]][b^2][a^1 b^3][b^1, c^1]]$
141.	$[[a^3, c^3][a^3, c^2][a^1 c^2]]$
142.	$[b^6, c^5]$

Fig. 3. State transition of ACS (λ_e in between 0.5 and 1.0)

The behavior of the ACS is classified by this parameter. When λ_e is small, cells do not evolve but instead disappear. As λ_e becomes larger, the ACS demonstrates “cell cycle”-like behavior, wherein a small cell grows larger before dividing into smaller cells, which then in turn grow larger and divide, and so on. When λ_e exceeds 1.0, the ACS grows larger and develops a complicated internal structure, but when it exceeds 2.5, since sufficiently large membrane compounds are being produced compared to the dissolving rate of membrane compounds, the cell does not divide and simply becomes a large cell with a simple structure.

This result shows that the consumption of membrane compounds results in dynamical behavior of the system, but while sufficiently large quantities of membrane compounds remain, the system is stable and does not demonstrate dynamical behavior.

Genetic ACS, GACS

In the previous simulation, we controlled the characteristics of reactions by changing the λ_e parameter. The next step was to use reaction rules to the control the ACS, specifically by introducing heredity of reaction rules: i.e., when a cell divides, the mutated reaction rules are inherited by the divided cell. In the GACS, we denote the set of reaction rules as the rule matrix in Table 1:

step	state
0.	$[a^{10}, b^{10}, c^{10}]$
1.	$[a^{10}, b^{10}, c^9]$
.....	
41.	$[a^2, b^7, c^5]$
42.	$[[b^2][b^6, c^1]]$
43.	$[[b^4][b^3][b^2, c^3]]$
44.	$[[b^4][b^2][b^2, c^1]]$
45.	$[[b^4, c^2][b^2, c^2]][b^1, c^3]]$
46.	$[[[b^3, c^1][b^1, c^1]][[b^2, c^1][b^1, c^3]]]$
47.	$[[[[b^2][a^1, b^1]][b^1, c^2]][a^1, b^3, c^2]]$
	$[a^1, b^2, c^3]]$
.....	
140.	$[[[a^1, b^1][b^1, c^1]][b^2][a^1, b^1][b^1, c^1]]]$
.....	
214.	$[[[[[a^{16}, b^3, c^2][a^5, b^5, c^1][a^3, b^2]]]$
	$[a^4, b^3]][[[[b^2][b^2][a^1, b^2]][a^3, b^1]]$
	$[a^3, b^1][b^4][a^3, b^1]]]$

Fig. 4. State transition of ACS (λ_e in between 1.05 and 2.33)

step	state
0.	$[a^{10}, b^{10}, c^{10}]$
1.	$[a^{12}, b^5, c^9]$
2.	$[a^{14}, b^4, c^{10}]$
3.	$[a^{13}, b^6, c^{11}]$
4.	$[a^{14}, b^8, c^{12}]$
5.	$[a^9, b^9, c^{10}]$
6.	$[a^7, b^{10}, c^{10}]$
.....	
87.	$[a^{17}, b^{12}, c^{21}]$
.....	
147.	$[a^{14}, b^{20}, c^{29}]$
.....	
242.	$[a^{17}, b^{50}, c^{44}]$
.....	
300.	$[a^3, b^{56}, c^{58}]$

Fig. 5. State transition of ACS (λ_e more than 2.5)

Table 1. Rule matrix

	a	b	c
a	x, y_{aa}	x, y_{ab}	x, y_{ac}
b	x, y_{ba}	x, y_{bb}	x, y_{bc}
c	x, y_{ca}	x, y_{cb}	x, y_{cc}

where x, y_{ij} represents the number of x compounds of i that are transformed from the number of y compounds of j ; when $x = 1$, it is not denoted. For example, $2, 3_{ab}$ is shorthand for $a, a \rightarrow b, b, b$. When the number on the left hand side of a rule (e.g., x) is one, we abbreviate it, so $a \rightarrow b, b, b$ is abbreviated as 3_{ab} .

Transmission of Reaction Rules. When a cell is divided, the reaction rules that govern the cell are copied and passed down to the new daughter cell. At that time, a point mutation occurs only in the copied rules passed down to the new cell; the initial cell retains its old rules. Point mutations occur at the cell division time point, rewrites x or y , and changes the number of transforming substances.

An Experimental Result of GACS

We set a GACS as $\Gamma = (A = \{a, b, c\}), R, \mu = \{0\}, M_0 = \{a^{10}, b^{10}, c^{10}\}, \delta = 0.4, \sigma = 0.2)$, where the R is defined as follows;

	a	b	c
a	0_{aa}	0_{ab}	1_{ac}
b	1_{ba}	0_{bb}	0_{bc}
c	0_{ca}	1_{cb}	0_{cc}

In the evolution of the GACS, we examined the productivity of the membrane compounds of R , which is defined as the ratio of the number of membrane compounds produced to the number of non-membrane compounds: for example, denotes $a \rightarrow b, b, c, c$ and $c \rightarrow a$. Two c are produced, while a combined three of a and b are also produced: hence, this ratio for c is $Prd_{mc} = \frac{2}{3}$.

	a	b	c
a	0_{aa}	0_{ab}	1_{ac}
b	2_{ba}	0_{bb}	0_{bc}
c	2_{ca}	0_{cb}	0_{cc}

At every step of the GACS, an acs (acs_i) is selected randomly from S_{acs} and m_i is rewritten using r_i ; the pre-defined fitness function $v_f(acs_i)$ gives the fitness value of acs_i . If this fitness value exceeds the required value θ , a sibling of acs_i is reproduced by mutating m_i and/or r_i . Otherwise, acs_i is removed from S_{acs} by dissolving its membrane to allow the internal compounds to be merged into the upper membrane, and a new acs , acs_i , is added to S_{acs} instead. The new acs_i has randomly generated reaction rules r_i and a randomly generated multiset m_i . An outline of the algorithm of the GACS is summarized below.

Figure6 illustrates the time series of productivity, where the vertical axis illustrates the steps and each dot is an R . It shows that initially, almost all R s evolve so that $Prd_{mc} > 1$. However, after 100 steps, the productivity of the R s decrease. At this point, both the number and size of cells increase exponentially.

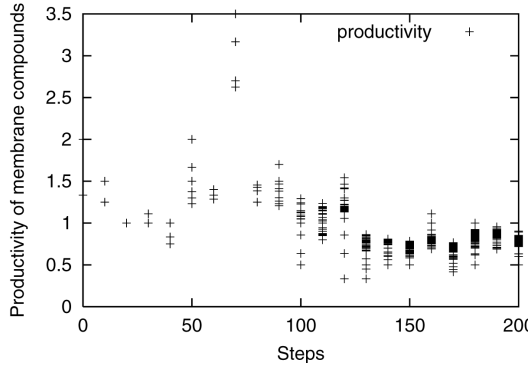


Fig. 6. Time series of productivity of membrane compounds

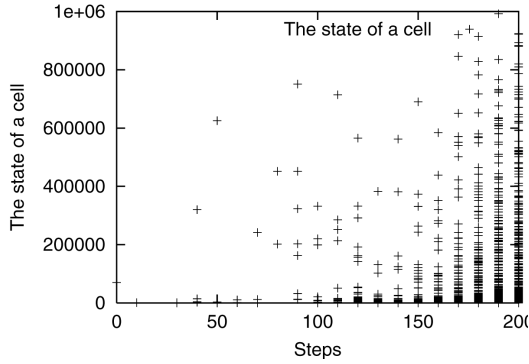


Fig. 7. Time series of the distribution of internal nodes of the whole system

Furthermore, the structure of the cells gains increasing complexity. Figure 6 illustrates the correlation between the number of cells, the size of the cells and the number of steps, where each dot corresponds to an individual cell. Figure 7 illustrates the internal nodes of the whole system. If we regard M_0 as the root and other cells as internal nodes and leaves, we can regard the whole system as a tree. We use the number of internal nodes in the tree as a metric of its complexity. Figure 7 illustrates that the number of internal nodes increases exponentially after 150 steps. It is interesting that when a cell grows into a hierarchical cell, the R evolves to have low productivity. The reason for this behavior could be that the R whose productivity is high always suffers from mutations, because it promotes membrane division and thus generates more mutations than low productivity R s. If every cell is an elementary cell, the R must keep producing membrane compounds at a high rate. However, when the cell forms internal structure, high productivity is not necessary, because in a structured cell, if an inside cell dissolves, the cell that includes the dissolved cell obtains its membrane compounds. Therefore, as a cell evolves into a structured cell, the cell

needs a high productivity of R . However, once it becomes a structured cell, high productivity of R is filtered out.

Modeling P53 Signaling Pathways

The p53 signaling network plays a major role in cell survival, as it safeguards against genetic instability, which can lead to tumor formation. However, the complicated structure of the network hampers modeling with ordinary rate equation models.

The p53 signaling network has been studied intensively because over 50 to 55% of all human cancers are reported to involve a mutation in the p53 gene [20]. The p53 protein is a transcription factor that plays a major role in regulating the response of mammalian cells to stresses and damage, mainly through the transcriptional activation of genes involved in cell cycle control (G1 arrest), DNA repair, and apoptosis [20]. In healthy cells, p53 is a short-lived and non-abundant protein, due to its rapid degradation [20]. However, in the presence of DNA damage, p53 transforms itself from a latent to an active conformation [20]. p53 has two levels of activation, depending on the level of DNA damage. Weakly activated p53 prevents damaged cells from proceeding in the cell division cycle and promotes DNA repair. Highly activated p53 induces apoptosis and eliminates mutated or irrevocably DNA-damaged cells.

To delay p53-induced apoptosis and permit cells that are not irreversibly damaged or mutated to survive, p53 forms an auto-regulatory negative feedback loop with MDM2 oncoproteins. In addition, the survival factor promotes the activation of MDM2 through PI3K-PDK1-Akt signaling and translocation of p53. Moreover, the growth factor inhibits MDM2 activation through PTEN protein and caspase activation, whereas P53 induces MDM2, which provides DNA-damaged cells the opportunity for DNA repair. Subsequently, p53 induces PTEN, which then induces the death of mutated or irrevocably DNA-damaged cells. P53 and MDM2 form a highly complex network which allows for the proliferation of healthy cells and the elimination of mutated cells [20].

In order to model these processes, we use an ACS with two membranes, one representing the nucleus and one enclosing the cytoplasm, which are labeled n and c , respectively. As such, the membrane structure is $[c[n]_n]_c$ where $[c]_c$ represents cytoplasm and $[n]_n$ represents for nucleus. The rewriting rules of p53 signaling network are given in Table2. Some rules require target membrane to be moved: for example, the rules for the “cytoplasm ($[c]_c$)“, $P53 \rightarrow (P53, Nucleus)$ means that a P53 in the cytoplasm will move to “nucleus ($[n]_n$)“ and instead of empty multiset, we have written “vanish” for clarity.

Biologically, MDM2 is phosphorylated by survival signaling through the PI3-kinase-PDK1-Akt pathway, which promotes rapid p53 degradation. We have summarized these interactions in the rule from R_n . Thus P53 and MDM2 form an auto-regulatory negative feedback loop. Once DNA damage increases P53 is activated and trans-located from the cytoplasm to the nucleus. The activated P53 complex induces PTEN protein and caspase activation that inhibit MDM2 activation. We have summarized these interactions in the rule

$$P53 - tetramer, DNA - damage \rightarrow P53 - tetramer^+, DNA - damage$$

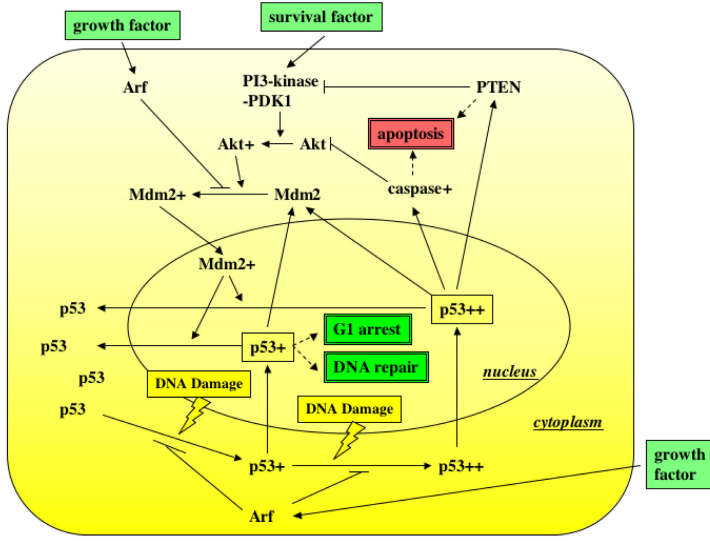


Fig. 8. Schematic model of P53 signal transduction system

Table 2. Rewriting rules of the P53 signaling network

Rules in <i>Cytoplasm</i> , R_c
$P53 \rightarrow \text{vanish},$
$P53 \rightarrow (P53, \text{Nucleus}),$
Rules in <i>Nucleus</i> , R_n
$P53, P53, P53, P53 \rightarrow P53\text{-tetramer},$
$P53 - \text{tetramer} \rightarrow P53 - \text{tetramer}, MDM2,$
$P53, MDM2 \rightarrow \text{vanish},$
$MDM2 \rightarrow (MDM2, \text{out}),$
$P53 - \text{tetramer}, MDM2 \rightarrow (P53, P53, P53, P53, \text{Cytoplasm}),$
$P53\text{-tetramer DND-damage} \rightarrow P53 - \text{tetramer}^+, \text{DNA-damage},$
$P53 - \text{tetramer}^+, \text{DNA-damage} \rightarrow P53\text{-tetramer}$
$P53, MDM2 \rightarrow \text{vanish}$

from R_n where “ $P53 - \text{tetramer}^+$ ” indicates that P53-tetramer has been activated. It forms the positive feedback loop to accelerate p53 activation. The results of the simulation correlate with biological data; when the DNA is damaged (the “abnormal state in Figure 9), a p53-tetramer is activated, which repairs DNA damage, before returning to the normal state, where it is degraded by MDM2 into single p53. Thus, p53 and MDM2 form an auto-regulatory negative feedback loop.

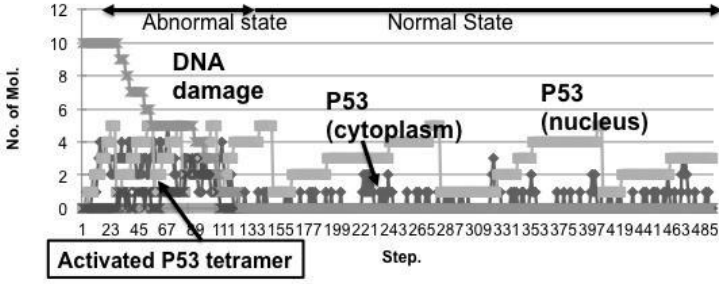


Fig. 9. Time series of concentration of P53 and MDM2 in P53 signaling network

4 Modeling Ecological System

We have modeled chemical reactions, membrane systems, and cell signaling networks. Lastly, we model a macro system - an ecosystem. An ecosystem has been reported in which plants may respond to herbivore feeding activities by producing volatile chemicals that attract carnivorous enemies of the herbivores [18]. These volatiles are not merely the result of mechanical damage, but are produced by the plant as a specific response to herbivore damage. We model this tritrophic system by using an ARMS with stochastic transition. Let the symbol “ a ” be a leaf, “ b ” be a herbivore, “ d ” be a carnivore and “ c ” be the density of herbivore-induced volatiles. Furthermore, we define “ e ” to be an empty state in order to introduce the death state. A plant is defined implicitly as a number of leaves. Evolution rule R_1 is defined as follows:

$$\begin{aligned}
 a &\xrightarrow{k_1} a, a \quad r_1 \quad (\text{increase in the number of leaves}), \\
 a, b &\xrightarrow{k_2} b, b, c \quad r_2 \quad (\text{herbivore eats a leaf}), \\
 d, b, c &\xrightarrow{k_3} d, d \quad r_3 \quad (\text{carnivore catches a herbivore}), \\
 d &\xrightarrow{k_4} e \quad r_4 \quad (\text{death of a carnivore}), \\
 b &\xrightarrow{k_5} e \quad r_5 \quad (\text{death of a herbivore}).
 \end{aligned}$$

Rule r_1 corresponds to the sprouting and growth of a plant, r_2 corresponds to a herbivore eating a leaf and the leaf generating volatiles, r_3 to the herbivore being preyed upon by a carnivore, r_4 to the death of a carnivore, and r_5 to the death of a herbivore. More precisely, r_2 denotes the case when a leaf (a) exists, and a herbivore (b) eats the leaf and reproduces. The leaf produces volatile compounds (c) upon being eaten, to attract carnivores. Rule r_3 denotes the case when there is a herbivore (b) present with volatiles (c), and a carnivore (d), attracted by the volatiles of r_2 , catches the herbivore and reproduces (d, d).

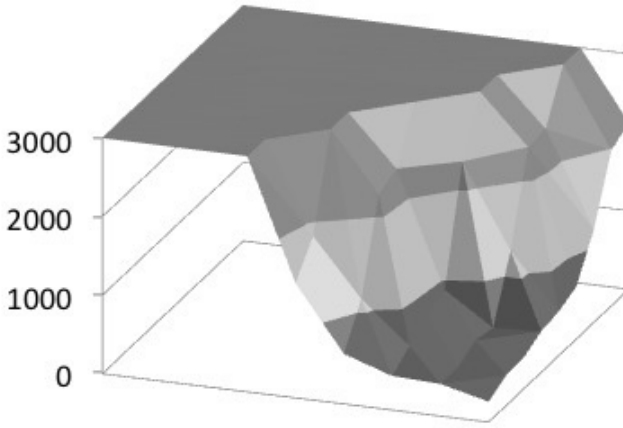


Fig. 10. Symbiotic relation of plant, herbivore and carnivores, horizontal axes illustrate k_1 and k_2 and the vertical axis illustrates step time of sustaining symbiotic relations; (with herbivore-induced plant volatiles HIPV)

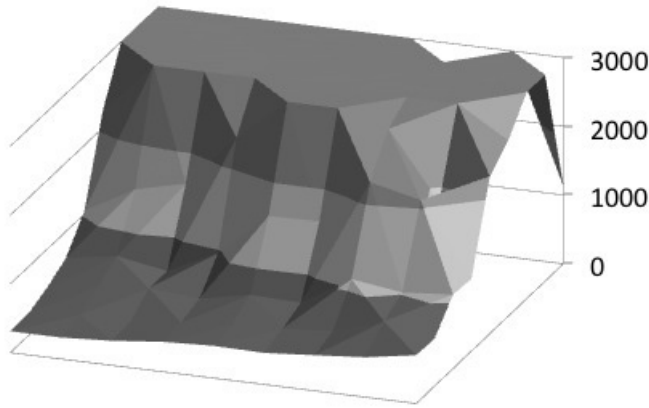


Fig. 11. Symbiotic relation of plant, herbivore and carnivores, horizontal axes illustrate k_6 and k_7 and the vertical axis illustrates step time of sustaining symbiotic relations; (without HIPV)

Using this model, we compared the outcomes of systems where leaves generate volatiles against those of systems where leaves do not. The evolution rules of the system without volatiles R_2 are defined as follows:

$$\begin{aligned}
 a &\xrightarrow{k_6} a, a \quad k_6 \quad (\text{increase in the number of leaves}), \\
 a, b &\xrightarrow{k_7} b, b \quad k_7 \quad (\text{herbivore eats a leaf}), \\
 d, b &\xrightarrow{k_8} d, d \quad k_8 \quad (\text{carnivore catches a herbivore}), \\
 d &\xrightarrow{k_9} e \quad k_9 \quad (\text{death of a carnivore}), \\
 b &\xrightarrow{k_{10}} e \quad k_{10} \quad (\text{death of a herbivore}).
 \end{aligned}$$

In both R_1 and R_2 , the symbiotic relationship between plants, herbivores and carnivores can be identified. It is interesting that R_2 (the relationship under a system without volatiles, Figure 11) is more likely to collapse than R_1 , under the same conditions, with the rate of collapse dependent on the number of plants, herbivores and carnivores in the initial state, and the reaction rate of r_3 and r_8 .

On the other hand, it shows that a system with volatile generation by plants (Figure 10) is more robust than a system without it under otherwise identical conditions. Thus, the role of volatiles warrants further investigation.

5 Dissipation of Information

By modeling self-organizing systems at such different scales—a BZ reaction, the p53 signaling system, and a chemical ecosystem—with ARMS, we have discovered that these models have a common algorithmic structure. In every model, interactions produce an “intermediate substance” that is consumed by evoking another interaction. In the BZ reaction, the intermediate product “Y” is produced by the reaction of “A, X → Y” and “Y” is consumed by evoking the auto-catalytic reaction “X, X, Y → X, X, X”. In the protocell system, membrane compounds “C” are produced by the reaction of “A, B → C” and dissipation of “C”, requiring the generation of another “C” in order to maintain the membrane (hence, dissipation evokes mobilization of “C”). In the p53 signaling system, the intermediate product “p53 – tetramer⁺” is produced by damaged DNA and consumed by repairing damaged DNA, a reaction that produces another intermediate product “p53-tetramer”, which in turn is consumed by generating a third intermediate product, “MDM2”, through the reaction of “p53-tetramer → p53-tetramer, MDM2”. In the ecosystem, volatile chemical “c” is produced by the feeding activity of herbivores, as described as the reaction of “a, b → b, b, c”, and is consumed by evoking the carnivore’s feeding on herbivores “d, b, c → d, d.”

Table 3. Dissipation of information in natural systems

BZ reaction	$X, X, Y \rightarrow X, X, X$
Proto cell	membrane compound → dissolved
P53 system	$P53 - tetramer^+, \text{DNA damage} \rightarrow P53 - tetramer^+$
Ecosystem	carnivore, herbivore, chemical → carnivore, carnivore

As summarized in Table 3, in natural systems, producing and consuming intermediate substances are important behaviors for controlling self-organization. These intermediate substances have not been prepared and are generated through interactions and consumed by evoking other reactions. As such, these intermediate substances can be regarded as “information”, and they regulate self-organization phenomenon. We call these information producing and consuming interactions “dissipation of information.” This is widely realized in natural systems, and so there are opportunities to manipulate natural systems through this mechanism.

6 Harness the Nature for Computation

Using this concept of dissipation of information, intermediate substances do not fully control the system, but only partly, and the internal dynamics of individual substances influence the dynamics of the whole system via a ripple effect. For example, in order to control a group of sheep, shepherds do not grip each individual sheep but threaten several sheep by voice or by using sheepdogs. This intimidation can be regarded as an “intermediate substance”: it is consumed through affecting the internal dynamics of a group of sheep, likely gathered together, as when one escaping sheep leads other sheep to follow. We refer to this indirect control of internal dynamics through affecting part of a system as “harness”, and those points where internal dynamics can be affected as “harness points.”

Harness a System by Dissipative of Information

We harness a food chain model which is composed of nine species of Lotoka-Volterra equations. Before harnessing, the model is denoted as;

$$\begin{array}{rcl} a_0, a_1 & \rightarrow & a_1, a_1, \\ a_1, a_2 & \rightarrow & a_2, a_2, \\ & \dots\dots\dots & \\ a_9, a_0 & \rightarrow & a_0, a_0, \end{array}$$

We set initial species populations as 20, 30, 40, 50, 60, 70, 80, 90, 100, then ran the system through 190 time steps. Species whose initial populations were 90, 70, 50, and 30 become extinct (Figure6). We harness the system by using

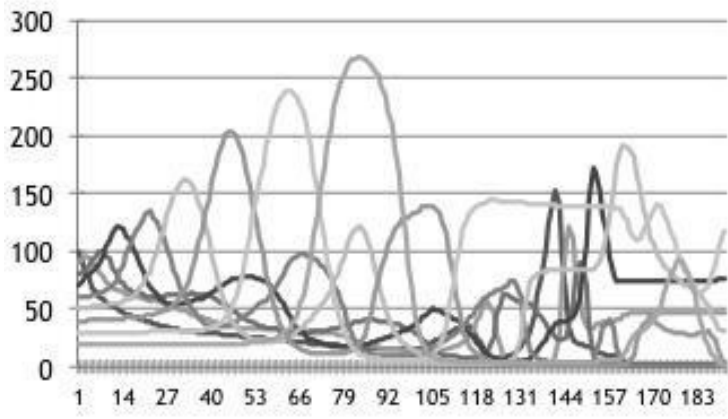


Fig.12. Population dynamics of the food chain which is composed of 10 species of Lotoka-Volterra model

information dissipative through introducing intermediate substances c_0, \dots, c_9 as harness points;

$$\begin{aligned}
 a_0, a_1, c_0 &\rightarrow a_1, a_1, c_1, \\
 a_1, a_2, c_1 &\rightarrow a_2, a_2, c_2, \\
 &\dots\dots\dots \\
 a_9, a_0, c_9 &\rightarrow a_0, a_0, c_0,
 \end{aligned}$$

where at each food chain step, an intermediate substance is consumed and produced, hence oscillations become very small (Figure 6).

After introducing harness points, although the distribution of initial populations remained the same, large amplitude oscillations were inhibited and no species become extinct (Figure6).

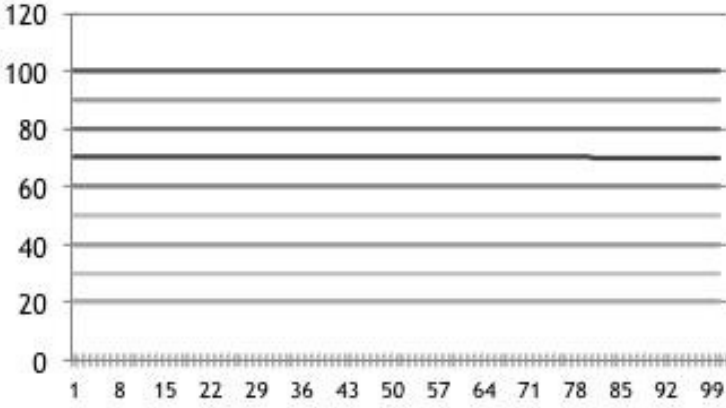


Fig. 13. Harnessed population dynamics of the food chain which is composed of 10 species of Lotoka-Volterra model with intermediate substances. It is noticed that, scale of the vertical axis is different from Figure6.

GPA

We harness GACS to solve a problem and named it the Genetic Protocell Algorithm (GPA). We solve the *doubling problem*, calculating the sum of a and b then showing the result as a number of c ($c = 2(a + b)$); The construct of GPA Γ^{GPA} is

$$\Gamma^{GPA} = (S_{acs}, f_s, M_s, \theta),$$

where

- (1) $S_{acs} = \{acs_1, \dots, acs_n\}$ is a set of ACS;
- (2) $f_s(acs_i)$ is a function which gives the fitness value of acs_i ;
- (3) $M_s(acs_i)$ is a operation which mutates acs_i ;
- (4) θ is the threshold of fitness value for natural selection;

In the initial state, we set 100 elementary cells inside M_0 . No compounds are transformed among protocells and no input and output are assumed.

The way of dissolving and dividing are the same as in GACS. After n rewriting steps, if the number of c in an acs is smaller than 7, its membrane is dissolved (a harness point), while if the number of c is larger than 9, the membrane is divided and a point mutation R is passed down to the new membrane. When a membrane is divided, chemicals inside the divided cell and its parent cell are reset to $\{a^2, b^2, c^0\}$ and the same problem again is solved again. The algorithm of GPA is summarized as follows;

```

begin
while (flag = NULL)
  select  $acs_i \in S_{acs}$  randomly
  rewriting  $m_i \in asc_i$ 
  obtain the fitness value  $v_f = f_s(m_i)$ 
  if ( $v_f < \theta$ )
    then
       $S_{acs} - acs_i$ 
      generate a new  $acs_i$  randomly
       $S_{acs} \cup acs_i$ 
    else
      a generate new acs,  $acs_j$  by mutating
       $M \cup acs_j$ 
  if  $\exists acs_k \in S_{acs}$  convergence
    then flag:=T
end.

```

An Experimental Result of a GPA

The GPA in this experiment was defined as follows; $\Gamma^{GPA} = (S_{acs}, f_s = M(c), c \in acs_i (1 \geq i \geq 100), \theta = f_s(acs_i) < 7)$, where $acs_i (1 \geq i \geq 100)$ is defined as follows;

$$\Gamma = (A = \{a, b, c\}, \mu, \mu = \{[,]_0, \dots [,]_{100}\}, M_0 = \{[a^2, b^2, c^0]^{100}\}, R, MC = \{\})$$

and θ is the same as previous section and σ is not used. In the initial state, R is After 5,000 reaction steps, every R which reached the solution within 8 steps was

	a	b	c
a	0_{aa}	0_{ab}	1_{ac}
b	1_{ba}	0_{bb}	0_{bc}
c	0_{ca}	1_{cb}	0_{cc}

selected; Chemical components of selected cells were reset to the initial state, $[a^2, b^2, c^0]$ and the calculations were performed again. Next, R s that solved the problem within 5 steps were selected. In this case, they had converged into the same reaction rules as follows; this R is closed to the result; By harnessing GACS,

we attempted to treat a system as a living thing, whereby in order to obtain the desired result, we observed its output (behaviors) and harnessed them by changing the environment without halting computation; by harnessing, natural selections and mutations lead to the GACS that achieves our goal.

$$\begin{array}{ll}
 a \rightarrow a,c, & a \rightarrow a,b,c,c, \\
 b \rightarrow a,b,c,c, & b \rightarrow a,b,c,c, \\
 c \rightarrow c,c; & c \rightarrow c,c; \\
 & a \rightarrow a,c,c, \\
 & b \rightarrow b \\
 & c \rightarrow c,c; \\
 \\
 & a \rightarrow a,c,c, \\
 & b \rightarrow c,c, \\
 & c \rightarrow c,c;
 \end{array}$$

While the concept of a harness has the concept been used, but not named as a “harness,” its realizations have already been used in various fields, as described in the following examples.

Spray Treatment. Secretory otitis media (SOM), persistent fluid in the middle ear cavity, stems from an unknown cause. S Skovbjerg et al. [12] used a nasal spray containing alpha-streptococcal bacteria considered to have a protective effect for children prone to middle ear infections, or otitis media. They investigated the clinical, bacteriological, and immunological effects of treatment with pro-biotic bacteria on SOM.

In the double-blind pilot/preliminary study, 60 children with chronic SOM (median 6 months) scheduled for insertion of tympanostomy tubes were assigned randomized nasal spray treatments with *Streptococcus sanguinis*, *Lactobacillus rhamnosus*, or placebo for 10 days before surgery. Clinical evaluation was carried out after 10 days of treatment. Middle ear fluid (MEF) was collected during surgery for quantification of cytokines and detection of bacteria by culture and polymerase chain reaction. Nasopharyngeal swabs were obtained before treatment and at surgery.

Complete or significant clinical recovery occurred in 7/19 patients treated with *S. sanguinis* compared to 1/17 patients in the placebo group ($p < 0.05$). In the *L. rhamnosus* treatment group, 3/18 patients were cured or became much better ($p = 0.60$; compared to placebo). Spray treatment did not alter the composition of the nasopharyngeal flora or the cytokine pattern observed in the nasopharynx or MEF, except for a higher level of IL-8 found in the nasopharynx of *L. rhamnosus*-treated children. This study shows that spray treatment with *S. sanguinis* may be effective against SOM.

Harness Point. In the spray treatment, the internal dynamics is the population dynamics between the causative microorganism and the resident microbiota and a harness point is to support the resident microbiota by the nasal spray.

$$\begin{aligned} a &\rightarrow c, c, \\ b &\rightarrow c, c, \\ c &\rightarrow c. \end{aligned}$$

Pest Control Technique. In the previous section, we modeled an ecosystem where plants responded to herbivore feeding activity by producing volatiles that attracted carnivorous enemies. J. Takabayashi et al. have used this model as harness for a pest control technique [22]. They found that cruciferous plants infested by diamondback moth larvae emitted a blend of volatiles that attract *Cotesia vestalis*, carnivorous braconid wasps that are natural enemies of the larvae. They also conducted experiments using a synthetic blend of *C. vestalis* attractants for diamondback moth control. They have identified the braconid inducement material triggered by the diamondback moth, and have developed an action control technique for the attracted wasp and another technique for activating the induced wasp.

Harness Point. In the pest control, the internal dynamics is the information dissipative in the population dynamics of ecosystem and a harness point is intermediate substance, chemical volatiles, HIPV.

7 Conclusion

We have considered the harness from the viewpoint of natural computing. We investigated facsimile computational models of spontaneous self-organization in nature, and identified dissipation of information flow as a common mechanism. Since the dissipation of information algorithm only partly affects a system’s internal dynamics, we consider it an indirect method of control, and conceived the concept of harnessing natural systems for the harness. We discovered that this concept is already used in harnessing as summarized in Table 4.

Table 4. Examples of different kinds of harness based on the concept of a harness, classified according to synthesis method, harness point, and materials

class	Primary	Harness point	Secondary
Wetware	Microbiology	population dynamics	Spray treatment
	Chemical Ecology	Dissipation of Info.	Pest Control
Software	Simulation of	Dissipation of Info.	Genetic Proto-cell
	Proto-cell	and population dynamics	Algorithm

Roughly, we can have two types of assumptions of the world: the digital world assumption (DWA) and the analog world assumption (AWA). In the DWA, every substance has a digital existence (i.e., either exists or does not exist) while in the AWA, every substance has an analog existence. Hence, in the DWA, we can use proof by contradiction, a powerful and convenient tool for deduction; however, we cannot use it in the AWA. Computing, although usable in the DWA, is an even more important tool within the AWA.

Nature is composed of gray boxes: in many cases, we can partly know about what is inside a box but not completely know about it. We describe a world composed of gray boxes as the oracle world, and a world composed of white boxes as the constructive world.

In computer science, we have mainly designed algorithms in the constructive world with the DWA; however, we can not compose algorithm by using gray boxes but white boxes. Hence we propose a constructive method as harness; science for the harness requires control methods in the oracle world with the AWA. Natural computing will be able to offer this through the paradigm of harnessing nature for computing.

In this position paper, we propose the concept of a harness and show that it has already been utilized for creating the harness. In order to better understand the harness concept, basic principles must be identified by idealizing it. Harnessing has nonetheless been used heuristically, and here we have shown examples of harnessing through dissipative information, a technique that could be used for multiscale phenomena such as drug design, environment conservation, and consensus making in a society.

Acknowledgments. This work is supported by JSPS KAKENHI Grant Numbers 23300317 and 24520106 and the Grant-in-Aid for Scientific Research on Innovative Areas Grand Number 2404002.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution Noncommercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Dittrich, P., Ziegler, J., Banzhaf, W.: Artificial chemistries, a review. *Artif. Life* 7(3), 225–275 (2001)
2. Jain, S., Krishna, S.: A model for the emergence of cooperation, interdependence, and structure in evolving networks. *Proc. Nat. Acad. Sci.* 98(2), 543–547 (2001)
3. Field, R.J., Burger, M.: *Oscillation and Traveling Waves in Chemical Systems*. Wiley, New York (1985)
4. Ganti, T.: Organization of chemical reactions into dividing and metabolizing units: the chemotons. *BioSystems* 7, 15–21 (1975)
5. Gillespie, D.T.: A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions. *J. Comp. Phys.* 22, 403–434 (1976)
6. Kakimoto, K., Taura, T.: A framework for analyzing sustainability by using the rewriting system. In: *Pro. Intern. Symp. on Environmentally Conscious Design and Inverse Manufacturing*, vol. 3, pp. 69–74. IEEE (2003)
7. Kakimoto, K., Shiose, T., Taura, T.: Analyzing Sustainability of Circulatory System by Using the Rewriting System. In: *Proceedings of EcoDesign: 2nd International Symposium on Environmental Conscious Design and Inverse Manufacturing*, pp. S103–S107. IEEE (2001)

8. Luisi, P.L.: Defining the transition to life: self-replicating bounded structures and chemical autopoiesis. In: Stein, W., Varela, F. (eds.) *Thinking About Biology*. Addison-Wesley, New York (1993)
9. Luisi, P.L., Varela, F.: Self-replicating micelles: a chemical version of minimal autopoietic systems. *Origins Life Evol. Biosphere* 19, 633–643 (1989)
10. *International Journal of Natural Computing*. Springer
11. Nicolis, G., Prigogine, I.: *Exploring Complexity, An Introduction*. Freeman and Company, San Francisco (1989)
12. Skovbjerg, S., Roos, K., Holm, S.E., et al.: Spray bacteriotherapy decreases middle ear fluid in children with secretory otitis media. *Arch Dis. Child* (August 19, 2008), doi:10.1136/adc.2008.137414
13. Suzuki, H.: An Approach to Biological Computation: Unicellular Core-Memory Creatures Evolved Using Genetic Algorithms. *Artificial Life* 5(4), 367–386 (2000)
14. Suzuki, Y., Tsumoto, S., Tanaka, H.: Analysis of Cycles in Symbolic Chemical System based on Abstract Rewriting System on Multisets. In: *Proceedings of International Conference on Artificial Life V*, pp. 482–489. MIT Press (1996)
15. Suzuki, Y., Tanaka, H.: Order parameter for symbolic chemical system. In: Adami, C., et al. (eds.) *Artificial Life IV*, pp. 130–142. MIT Press (1998)
16. Suzuki, Y., Tanaka, H.: Chemical evolution among artificial proto-cells. In: *Artificial Life VII*, pp. 54–64. MIT Press (2000)
17. Suzuki, Y., Fujiwara, Y., Takabayashi, J., Tanaka, H.: Artificial Life Applications of a Class of P Systems: Abstract Rewriting Systems on Multisets. In: Calude, C.S., Pun, G., Rozenberg, G., Salomaa, A. (eds.) *Multiset Processing*. LNCS, vol. 2235, pp. 299–346. Springer, Heidelberg (2001)
18. Suzuki, Y., Takabayashi, J., Tanaka, H.: Investigation of tritrophic interactions in an ecosystem using abstract chemistry. *J. Artif. and Robot.* 6(3), 219–223 (2001)
19. Suzuki, Y., Davis, P., Tanaka, H.: Emergence of auto-catalytic structure in stochastic self-reinforcing reaction networks. *J. Artif. and Robot.* 7, 210–213 (2003)
20. Suzuki, Y., Tanaka, H.: Modeling P53 signaling network by using multiset processing. *Applications of Membrane Computing Series: Natural Computing Series*, pp. 203–215. Springer, Tokyo (2006)
21. Suzuki, Y.: An investigation of the Brusselator on the mesoscopic scale *Inter. J. of Parallel, Emergent and Distributed Sys.* 22(2), 91–102 (2007)
22. Urano, S., Uefune, M., Takabayashi, J.: Inspection example of the diamondback moth prevention - effect with natural enemy attractant “bee cool”. In: 18th, Special Interest Group meeting of Natural Enemy Usage, The Japanese Society of Applied Entomology & Zoology (2008)
23. Walde, P., Goto, A., Monnard, P.A., Wessicken, M., Luisi, P.L.: Oparin’s reactions revisited: enzymatic synthesis of poly(adenylic acid) in micelles and self-reproducing vesicles. *J. Am. Chem. Soc.* 116, 7541 (1994b)